R Notebook for: Matrix Algebra Practice

Neil Yetz & Gemma Wallace

Contents


```
#Clear environment
rm(list=ls())
```

Helpful matrix algebra resources:

https://www.mathsisfun.com/algebra/matrix-multiplying.html

https://www.statmethods.net/advstats/matrix.html

Relevant syntax: ?hist %*% is the command for multiplying two matrices

matrix() is the function for creating a matrix

t() is the function for transposing matrices

Create matrices

Create a 3x3 matrix

This represents R in multivariate corrlations (the original correlation coefficients) Y1 Y2 Y3 X1 1.00 0.75 0.50 X2 0.85 1.00 0.30 X3 0.60 0.35 1.000

```
cor_matrix <- matrix(c(
  1.00,.75,.50,
  .85,1.00,.30,
  .60,.35,1.00),
  ncol=3,
  byrow=T)

cor_matrix
```

```
##      [,1] [,2] [,3]
## [1,] 1.00 0.75  0.5
## [2,] 0.85 1.00  0.3
## [3,] 0.60 0.35  1.0
```

Create a 1x3 matrix

This will represent weights (w) in multivariate correlation Y1 Y2 Y3 X1 1.0 1.0 1.0

```
weights <- matrix(c(1.0,1.0, 1.0),ncol=3,byrow=T)
weights
```

```
##      [,1] [,2] [,3]
## [1,]    1    1    1
```

Transpose the 1x3 matrix

This will represent w' (the transposed weights) in multivariate correlation

```
weights_t <- t(weights)
weights_t
```

```
##      [,1]
## [1,]    1
## [2,]    1
## [3,]    1
```

Multiply matrices

Multiply the 3x3 and 1x3 matrices

This will represent wR in multivariate correlation (the product of the original correlation matrix and the assigned weights)

```
wR_matrix <- weights %*% cor_matrix
wR_matrix
```

```
##      [,1] [,2] [,3]
## [1,] 2.45  2.1  1.8
```

```
# Note: Order matters in matrix multiplication! The number of columns of the left matrix must be the sa
```

Multiply this new matrix with the transposed 3x1 matrix

This will represent wRw' in multivariate correlation (the product of the original correlation matrix, the weights, and the tranposed weights). Since all of the values in the 1x3 and 3x1 matrices were equal to one (i.e., all variables equally weighted), this is the simple sum of all of the elements in the matrix.

```
wR_wprime_matrix <-  wR_matrix %*% weights_t
wR_wprime_matrix
```

```
##      [,1]
## [1,] 6.35
```

```
# Again, order matters! To get the sum of all of the elements in the matrix, put the wR_matrix first.
```